OFS -> Oracle file system (based on linux kernel)

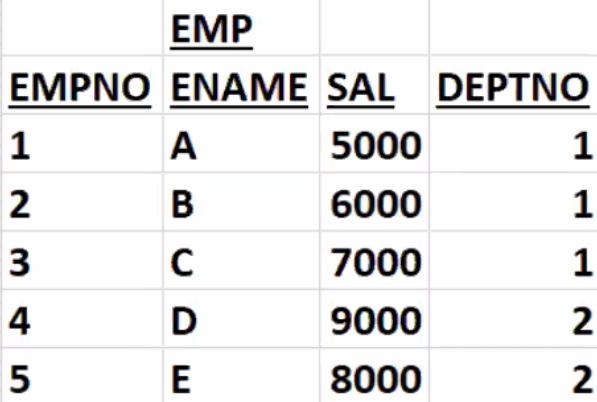
Oracle linux (known as unbreakable linux) -> most secure linux in the world (designed specifically to host the database)

MySQL – STORED OBJECTS

* Objects that are stored in the database
* E.g. tables, indexes

**VIEWS (VERY IMPORTANT)**

* Present in all RDBMS and some of the DBMS also



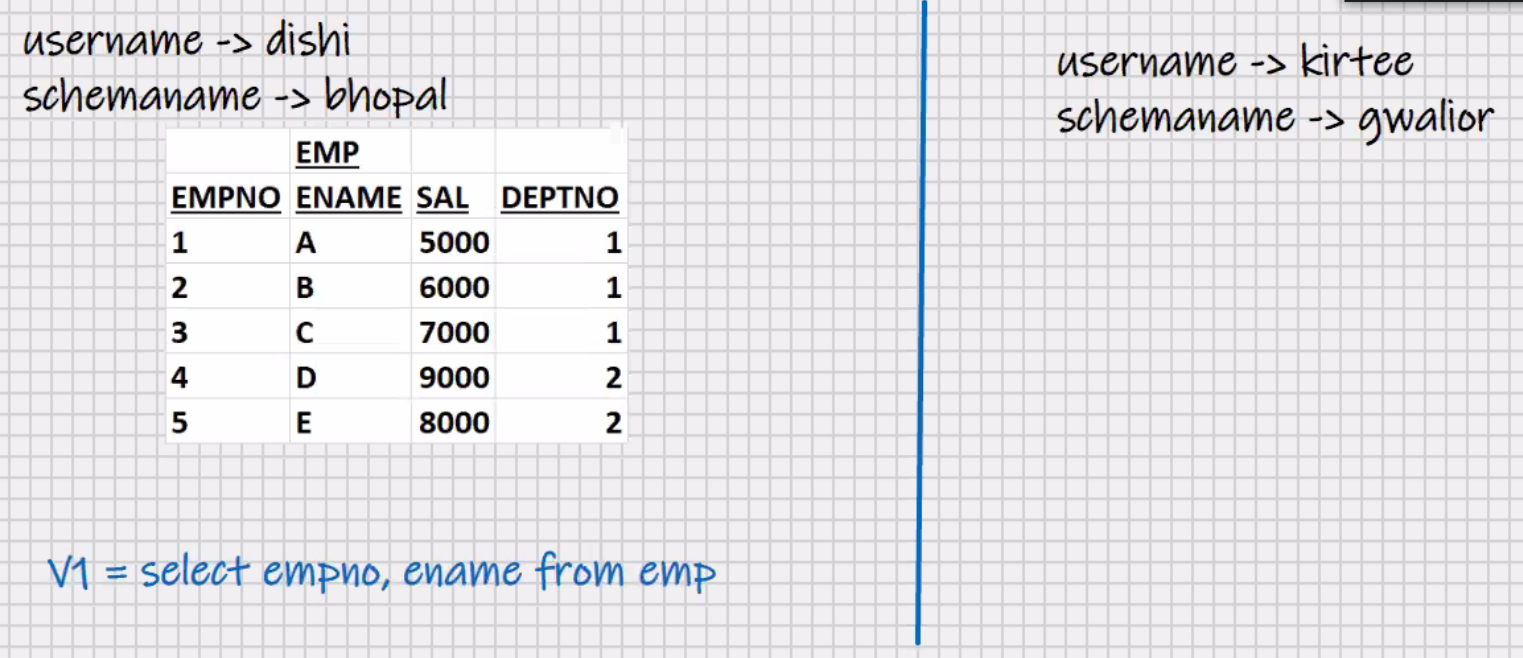
* View is a handle to a table
* View is a HD pointer(stores the address of table)
* View is known as locator (locator is a HD pointer)
* Used to restrict the access of users
* For **security purposes**
* Used for indirect access to the table

dishi\_mysql > create view v1

as

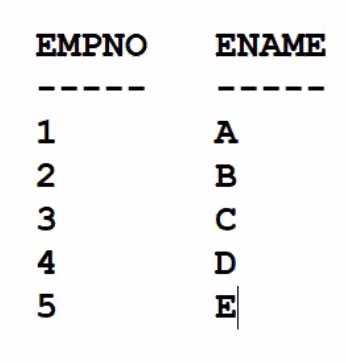
select empno, ename from emp;

View created.



Dishi\_mysql> grant select on v1 to kirtee;

Kirtee\_mysql> select \* from Bhopal.emp; -> error

Kirtee\_mysql> select \* from Bhopal.v1;

* Used to restrict the column access
* View is a form of Encapsulation

(form of data hiding)

Dishi\_mysql> grant select on v1 to

kirtee,Vaibhav,manoj;

* View does not contain data
* Only the definition is stored, data

Is not stored

* View is a stored query
* The select statement on which is based, is stored in a system table
* The select statement on which is based is stored in the DB in the compiled format
* View is an executable format of select statement
* Hence the execution is very fast
* Hiding the source code from end user

Kirtee\_mysql> insert into Bhopal.v1 values(6,’F’);

* DML operations can be performed on a view
* DML operations can be done on a view will affect the base table
* Constraints that are specified on the base table, they will be enforced when you INSERT via the view
* ENTIRE APPLICATION IS BUILT ON VIEWS

Dishi\_mysql> drop view v1;

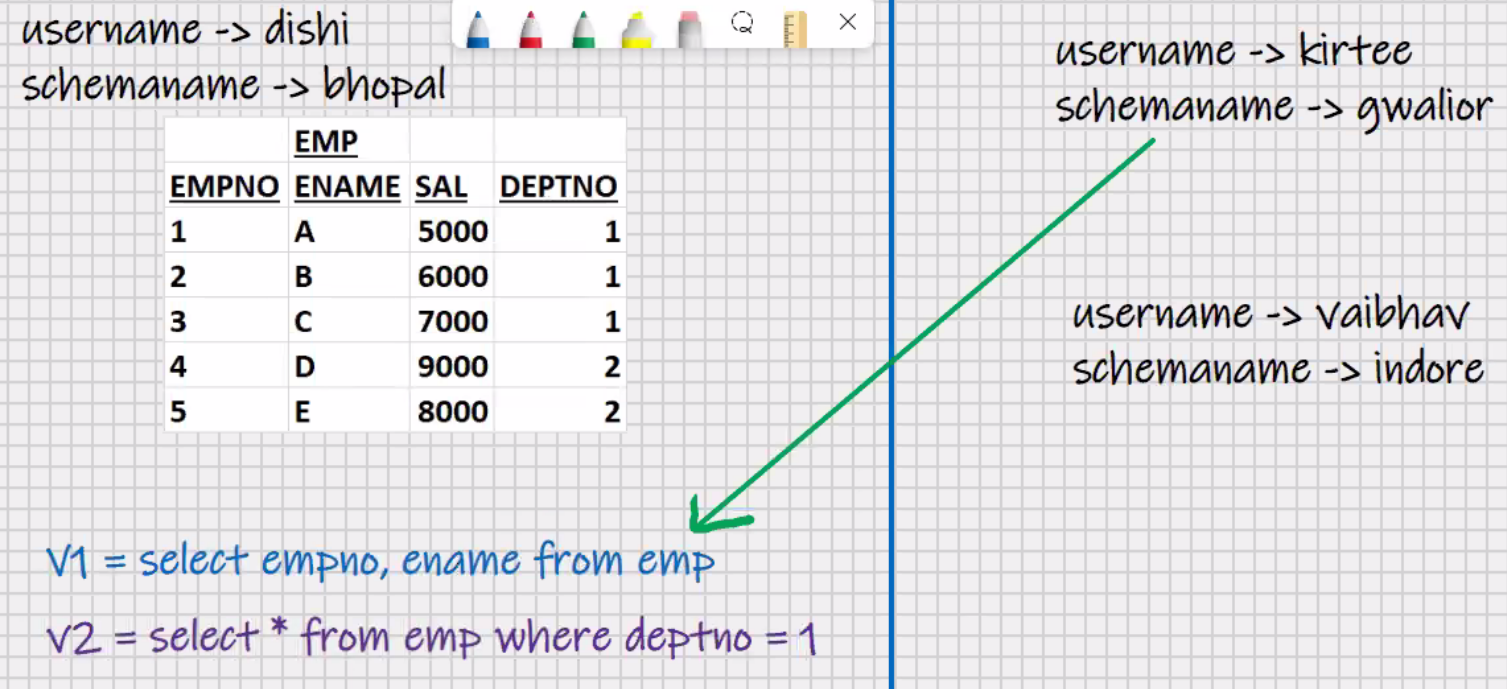
* Only owner can drop the view

Dishi\_mysql> create view v2

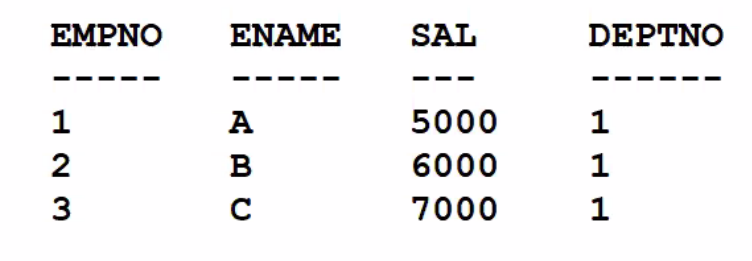
As

Select \* from emp where deptno = 1;

*View created.*



Dishi\_mysql> grant select on v2 to Vaibhav;

Vaibhav\_mysql> select \* from Bhopal.v2;

* Used to restrict the row access

Dishi\_mysql> grant select, insert on v2 to Vaibhav;

Vaibhav\_mysql> insert into Bhopal.v2 values(6, ‘F’,6000, **2)**;

->**allowed**

Dishi\_mysql> create view v2

As

Select \* from emp where deptno = 1 with check option;

Dishi\_mysql> grant select, insert on v2 to Vaibhav;

Vaibhav\_mysql> insert into Bhopal.v2 values(6,’F’,6000,**2**); **<- error**

* View with check option is similar to check constraint
* Used to enforce different checks for different users

Dishi\_mysql> Drop view v2;

Create view v1

As

Select empno, ename from emp;

To change the select statement in which the view is based:-

Create view v1 as ………. ;

**OR**

**Create or replace** view v1 as select ………. ;

Create or replace view v1

As

Select ename, sal from emp;

Select \* from v1;

* View based on computed column, expression, function, etc
* You can only select from this view
* DML operations are not allowed

Create or replace view v1

As

Select upper(ename) u\_ename, sal\*12 annual from emp;

Create or replace view v1

As

Select deptno, sum(sal) from emp

Group by deptno;

* View based on **group by clause**
* You can only select from this view
* DML(insert,update,delete) operations are not allowed

Create or replace view v1

As

Select dname,ename from emp,dept

Where dept.deptno = emp.deptno;

Select \* from v1;

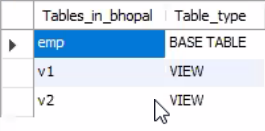
* View based on **join**
* You can only select from this view
* DML(insert,update,delete) operations are not allowed
* View based on view is allowed
* Uses

1. Union of>255 select statements
2. Function within function > 255 levels
3. Sub-queries > 255 levels
4. To simplify the writing of complex select statements

e.g. join of 40 tables, etc.

* Complex queries can be stored in view definition

show tables; -> will show tables and views but it wont tell which is view

to see which is a table and which is a view

show full tables;

to select statement on which the view is based:-

show create view v1;

use information\_schema;

show tables;

show full tables;

show create view character\_set; //cant see source code -> error

Advanced features:-

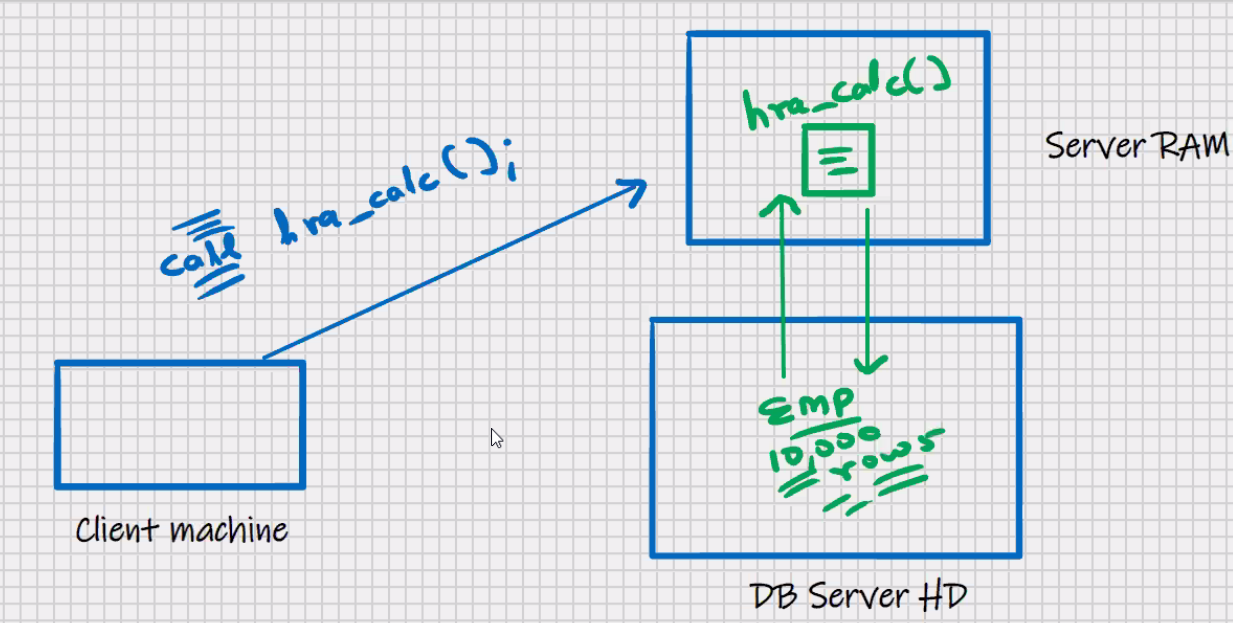
* Types of views
* Uses:-
* To enforce relational methods on object tables
* To enforce object method on relational tables
* Convert 3D table into 2D table
* Convert 2D table into 3D table
* Migration
* Data mapping
* EAI (enterprise application integration)
* ElM (enterprise integration management)

**MySQL -PL**

* MySQL programming language
* Programming language of MySQL
* Used for database programming

E.g. HRA\_calc, TAX\_calc, Attendance\_calc, etc.

* Used for server-side data processing



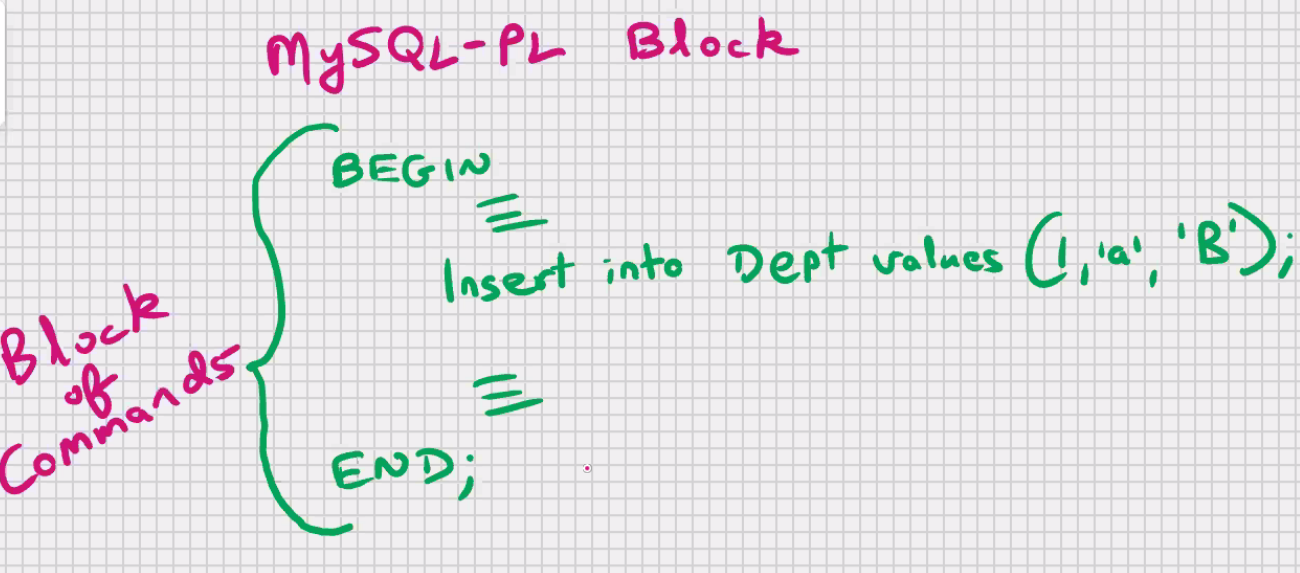
* MySQL -PL program can be called through MySQL command line client, MySQL workbench, oracle forms, oracle reports, oracle menus, oracle graphics, oracle apex(application express),java, ms .net, etc.
* Can be called trough any front-end s/w
* Few 4GL features

Mysql> call hra\_calc();

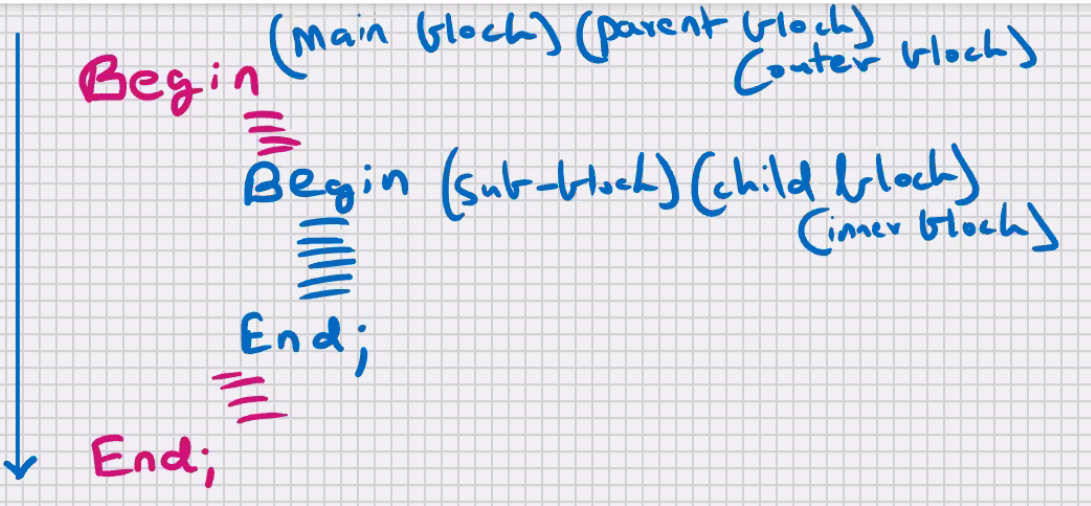
Begin

Insert into dept values(1,’a’,’B’);

End;



* Known as block level language
* Benefits of block within block



1. Modularity
2. Control scope of variables (form of data hiding)
   1. (form of Encapsula)
3. Efficient error management using exceptions

* Screen input and screen output is not allowed (scanf, printf not available)
* Used only for processing
* You can use select statement inside the block but its not recommended
* SQL commands that are allowed inside MySQL-PL:-

DDL,DML,DQL,DTL/TCL

* DCL commands are not inside MySQL-PL program

To store output of MySQL-PL program:-

Create table tempp

(

Fir int,

Sec char (15)

);

* MySQL-PL programs are written in the form of stored procedures

**STORED OBJECTS**

* Objects that are stored in the database
* E.g. create … tables, indexes, views
* Anything that you do with create command is a stored object

**STORED PROCEDURES**

* Routine (set of commands) that has to be called explicitly
* Global procedures
* Can be call MySQL command line client, MySQL workbench, java, MS.net etc.
* Can be called through any frontend s/w
* Stored in the database in the compiled format
* Execution will be very fast
* Hiding source code from end-user
* Execution takes place in server RAM
* Procedure can have local variables
* Within the procedure, you can have IF statement, loops, cursors, etc.
* One procedure can call another procedure (concept of calling procedure and called procedure)
* Procedure can call itself (known as recursion)
* You can pass parameters to a procedure (to make it flexible)
* Overloading of stored procedures is not allowed (you cannot create 2 or more procedures with the same name, even if the number of parameters passed is different, or the datatype of parameters passed is different)

Call abc ();

MySQL -> MySQL-PL

Oracle -> PL/SQL (procedural Language SQL)

MS SQL server -> T-SQL (Transact SQL)

**#1**

Delimiter //

create procedure abc ()

begin

insert into tempp values (1,'Hello');

commit;

end; //

delimiter;

call abc ();

select \* from tempp;

truncate table tempp;

🡪> read, compile, plan and store in the DB in the compiled format

Procedure created.

Drop procedure abc;

* ; is known as terminator (denotes end of command)
* ; is also known as delimiter

**#2**

Delimiter //

Create procedure abc ()

Begin

Declare x int;

Set x = 10;

Insert into tempp values (x, ‘Hello’);

End; //

Delimiter;

Call abc ();

Select \* from tempp;

* In MySQL if you declare the variable and you don’t initialize, then by **default it will store a null value**
* You can declare a variable and initialize it simultaneously

**#3**

Delimiter //

Create procedure abc ()

Begin

Declare x int default 10;

Insert into tempp values (x, ‘Hello’);

End; //

Delimiter;

Call abc ();

Select \* from tempp;

**#4**

Delimiter //

Create procedure abc ()

Begin

Declare x char (15) default ‘CDAC’;

Insert into tempp values (1, x);

End; //

Delimiter;

Call abc ();

Select \* from tempp;

**For char, varchar, date, time, datetime use ‘ ’ (single quotes)**

**#5**

Delimiter //

Create procedure abc ()

Begin

Declare x char(15) default ‘KING’;

Declare y float default 3000;

Declare z float default 0.4;

Declare hra float;

Set hra = y\*z;

Insert into tempp values (y, x);

Insert into tempp values (hra, ‘HRA’);

End; //

Delimiter;

Call abc ();

Select \* from tempp;

* In MySQL, float to int -> implicit conversion (auto-rounding takes place)

**#6**

Delimiter //

Create procedure abc (x char (15), y float, z float)

Begin

Declare hra float;

Set hra = y\*z;

Insert into tempp values (y, x);

Insert into tempp values (hra, ‘HRA’);

End; //

Delimiter;

Call abc(‘KING’,3000,0.4);

Call abc(‘SCOTT’,2500,0.3);

Select \* from tempp;

**#7**

Delimiter //

Create procedure abc (x char (15), y float, z float)

Begin

--Single line comment

/\* Multi line

comment\*/

End; //

Delimiter;

**Comments** -> internal documentation

* 1 comment min every 2 statements

Drop procedure abc;

**To see which all procedures you have created:-**

Show procedure status; <- show all procedure in all schemas

Show procedure status where db = ‘bhopal’;

Show procedure status where name like ‘a%’;

**To view the source code of stored procedure:-**

Show create procedure abc;

